

姓名：林婷婷

专业：信息管理与信息系统 班级：20181171

学号：2018117126

科目：Android 应用开发

实验日期：2021.6.3

实验题目：Activity

【实验目的】

探究 Activity

【实验环境及方式】

Android Studio

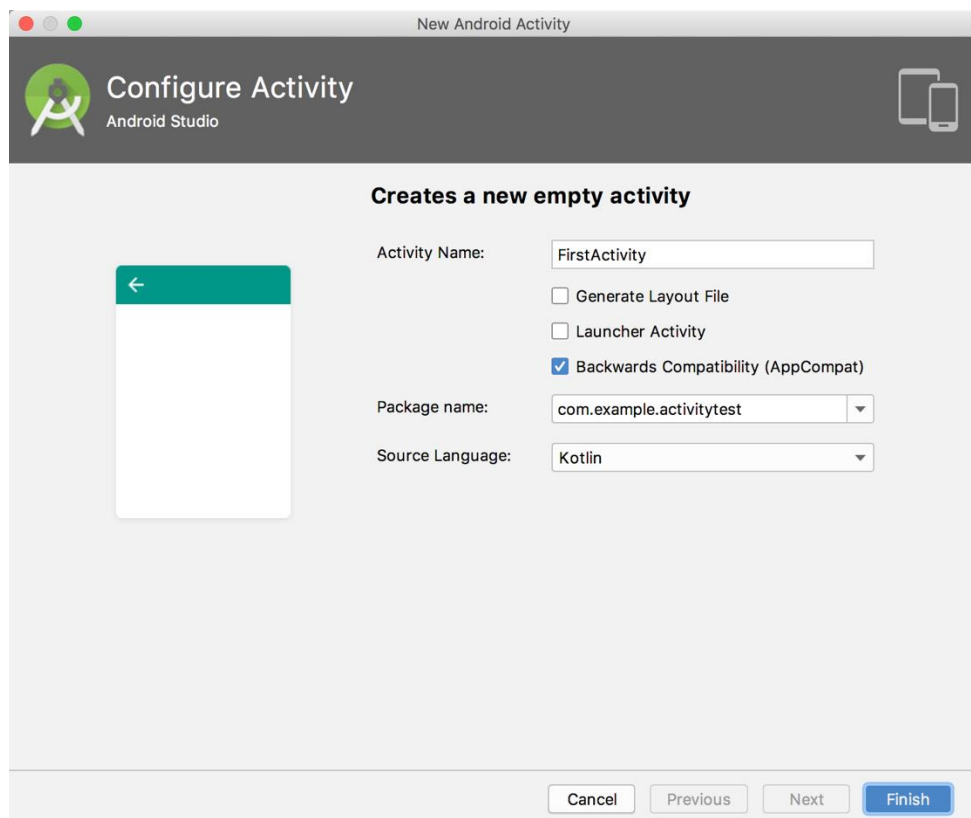
【实验内容及实验结果】

1. 什么是 Activity

Activity 是最容易吸引用户的地方，它是一种可以包含用户界面的组件，主要用于和用户进行交互。

2. 手动创建 Activity

右键项目的任何包路径→New→Activity→Empty Activity，会弹出一个创建 Activity 的对话框，如图所示。



① 选 Generate Layout File 表示会自动为 FirstActivity 创建一个对应的布局文件。

② 选 Launcher Activity 表示会自动将 FirstActivity 设置为当前项目的主 Activity。

韩山师范学院实验报告

③ 选 Backwards Compatibility 表示会为项目启用向下兼容旧版系统的模式。

④ 点击“Finish”完成创建。

项目中的任何 Activity 都应该重写 onCreate()方法，而目前 FirstActivity 中已经重写了这个方法，这是 Android Studio 自动帮我们完成的，代码如下所示：

```
class FirstActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
    }  
}
```

3. 创建和加载布局

Android 程序的设计讲究逻辑和视图分离，最好每一个 Activity 都能对应一个布局。

右击 app/src/main/res/layout→New→Directory，可以创建一个布局文件。布局文件中是使用 XML 来进行界面编辑的，这里创建一个 first_layout.xml 文件，并编写如下代码：

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
    <Button  
        android:id="@+id/button1"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:text="Button 1" />  
  
</LinearLayout>
```

回到 FirstActivity，在 onCreate()方法中加入如下代码即可给当前的 Activity 加载一个布局。

```
class FirstActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.first_layout)  
    }  
}
```

4. Activity 在 AndroidManifest.xml 中进行注册

所有的 Activity 都要在 AndroidManifest.xml 中进行注册才能生效。Activity 的注册声明要放在

韩山师范学院实验报告

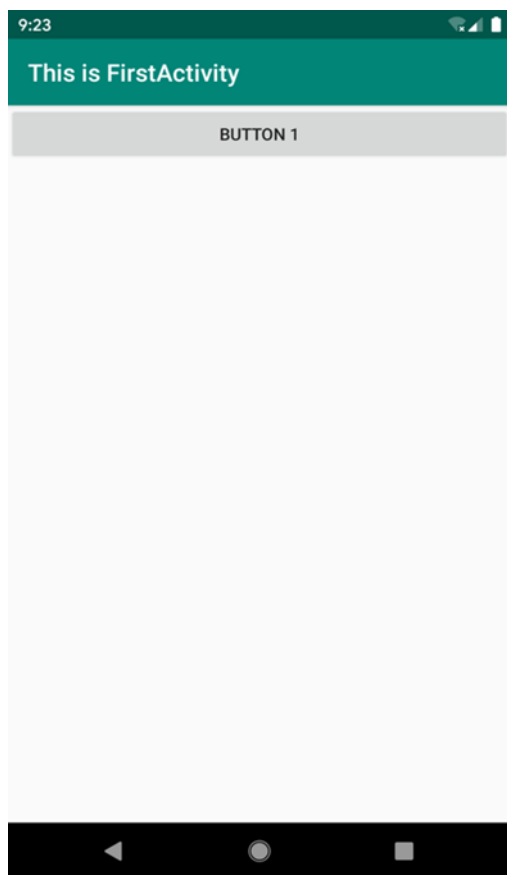
<application>标签内，并通过<activity>标签来对 Activity 进行注册。还可以通过在<activity>标签的内部加入<intent-filter>标签来配置主 Activity。

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.activitytest">

<application ...>
<activity android:name=".FirstActivity" android:label="This is FirstActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>

</manifest>
```

5. 程序的运行效果



6. 在 Activity 中使用 Toast

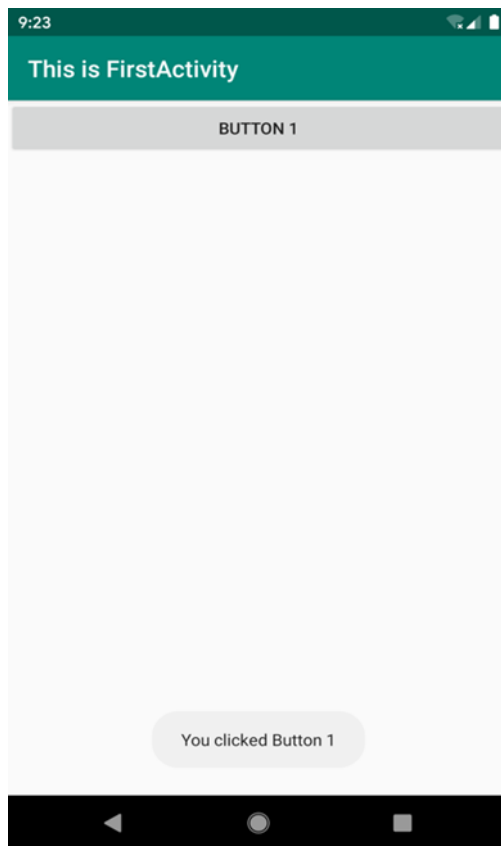
Toast 是 Android 系统提供的一种非常好的提醒方式。

韩山师范学院实验报告

在程序中使用它将一些短小的信息通知给用户，这些信息会在一段时间后自动消失，并且不会占用任何屏幕空间。

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.first_layout)  
    button1.setOnClickListener {  
        Toast.makeText(this, "You clicked Button 1", Toast.LENGTH_SHORT).show()  
    }  
}
```

运行效果如图所示：

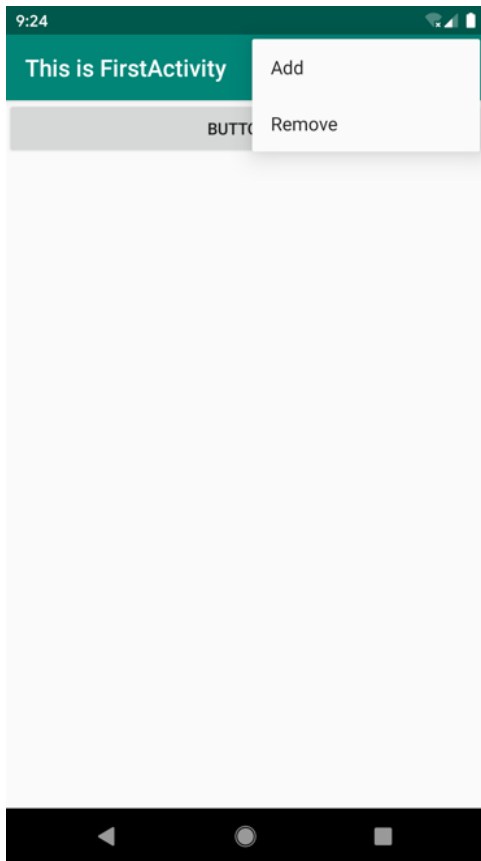


7. 在 Activity 中使用 Menu

Menu 提供了一种机制，可以让界面中的菜单项在默认情况下不显示。

只有当用户主动点击了菜单按钮时，才会弹出里面具体的内容，因此它不会占用任何 Activity 的空间。

```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {  
    menuInflater.inflate(R.menu.main, menu)  
    return true  
}
```



8. 销毁一个 Activity

方法一：按一下手机的 Back 键就可以销毁当前的 Activity

方法二：Activity 类提供了一个 finish() 方法，只需要调用一下这个方法就可以销毁当前的 Activity。

```
button1.setOnClickListener {  
    finish()  
}
```

Intent 的用法

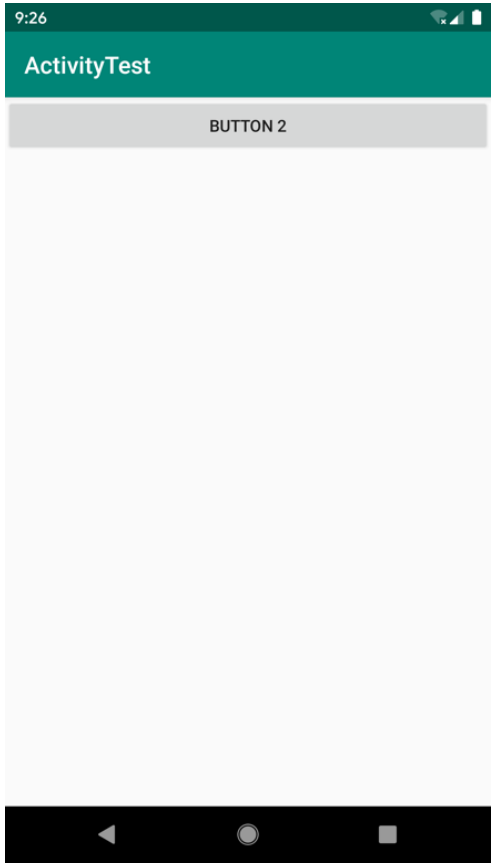
1. 使用显式 Intent

Intent 是 Android 程序中各组件之间进行交互的一种重要方式，它不仅可以指明当前组件想要执行的动作，还可以在不同组件之间传递数据。

显式 Intent 可以用于明确指定启动某个 Activity，代码如下所示：

```
button1.setOnClickListener {  
    val intent = Intent(FirstActivity.this, SecondActivity::class.java)  
    startActivity(intent)  
}
```

这段代码表示会从 FirstActivity 跳转到 SecondActivity。



2. 使用隐式 Intent

隐式 Intent 并不明确指出想要启动哪一个 Activity，而是指定了一系列更为抽象的 action 和 category 等信息，然后交由系统去分析这个 Intent，并帮我们找出合适的 Activity 去启动。比如你的应用程序中需要展示一个网页，这时没有必要自己去实现一个浏览器，只需要调用系统的浏览器来打开这个网页就行了。

```
button1.setOnClickListener {  
    val intent = Intent(Intent.ACTION_VIEW)  
    intent.data = Uri.parse("https://www.baidu.com")  
    startActivity(intent)  
}
```



韩山师范学院实验报告

3. Activity 的生命周期

onCreate() 这个方法在 Activity 第一次被创建的时候调用。

onStart() 这个方法在 Activity 由不可见变为可见的时候调用。

onResume() 这个方法在 Activity 准备好和用户进行交互的时候调用。

onPause() 这个方法在系统准备去启动或者恢复另一个 Activity 的时候调用。

onStop() 这个方法在 Activity 完全不可见的时候调用。

onDestroy() 这个方法在 Activity 被销毁之前调用。

onRestart() 这个方法在 Activity 由停止状态变为运行状态之前调用，也就是 Activity 被重新启动了。

【教师评语和成绩】

成绩：指导教师：日期：